Subject: explanation of c++ typedef line Posted by captainc on Thu, 18 Sep 2008 01:20:29 GMT View Forum Message <> Reply to Message

Can someone explain to me this line: typedef tagXYZHANDLE {} \* XYZHANDLE; My interpretation: 'XYZHANDLE is an alias of a pointer to an array of tagXYZHANDLE'? Is this correct?

It's from CodeProject article: HowTo: Export C++ classes from a DLL http://lamp.codeproject.com/KB/cpp/howto\_export\_cpp\_classes.aspx

Subject: Re: explanation of c++ typedef line Posted by mrjt on Thu, 18 Sep 2008 09:56:19 GMT View Forum Message <> Reply to Message

I don't know, but it doesn't compile. Perhaps an error?

Subject: Re: explanation of c++ typedef line Posted by cas\_ on Thu, 18 Sep 2008 12:27:10 GMT View Forum Message <> Reply to Message

captainc wrote on Thu, 18 September 2008 03:20 My interpretation: 'XYZHANDLE is an alias of a pointer to an array of tagXYZHANDLE'? Is this correct?

No, that would be:

typedef tagXYZHANDLE (\*XYZHANDLE)[];

Subject: Re: explanation of c++ typedef line Posted by captainc on Thu, 18 Sep 2008 13:23:34 GMT View Forum Message <> Reply to Message

This was one of the comments: Quote:typedef void\* XYZHANDLE; typedef void\* IJKHANDLE;

is not type safe, since all handle will became mutually replaceable being all void\*. (you can assign

an XYZ to an IJK).

One way to get around this is to declare a category (empty struct) and point to it.

typedef struct XYZHANDLE\_ {} \*XYZHANDLE; typedef struct IJKHANDLE\_ {} \*IJKHANDLE;

Now IJKHANDLE and XYZHANDLE cannot anymore be assigned each other.

Maybe the author replaced it incorrectly.

Subject: Re: explanation of c++ typedef line Posted by mr\_ped on Thu, 18 Sep 2008 14:27:53 GMT View Forum Message <> Reply to Message

There are more uncompilable things: #include "XyzLibrary.h" extern "C" XYZAPI IXyz\* APIENTRY GetXyz(); (i.e. it's probably problem with quality of the article itself)

That line from first post really looks like missing "struct" keyword, that was also my first idea without reading the article.

Page 2 of 2 ---- Generated from U++ Forum